AHDB1 Manuscript 1 (Sex-specifc Mortality) Redo (Heidinger)

Brynleigh Payne

2025-08-04

Prepare the data for analysis

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Import qPCR output for each plate 1) the Single Copy Autosomal Gene (EEF2) and mtDNA gene multiplex and 2) the Telomere reaction.

```
SCNAG <- read.csv("AHDB1_Manuscript1_Deaths_Multiplex_2025-08-01 13-38-10_795BR20744 -  Quantification
dim(SCNAG)
```

```
## [1] 150  16
```

```
Telo <- read.csv("AHDB1_Manuscript1_Deaths_Telomere_Heidinger_2025-08-01 11-52-51_795BR20744 -  Quantifi
dim(Telo)
```

```
## [1] 75 16
```

Edit

```
# Concatenate data across runs for the same samples
Plate1 <- rbind(SCNAG, Telo)
dim(Plate1)
```

```
## [1] 225  16
```

```
# Add a column called PlateID and fill in correct Plate number to use as a variable in the statistics
Plate1$PlateID <- "Plate1"

# CHECK AND EDIT FOR YOUR DATA.
#Name Correct Targets based on the fluorphores used in your reaction.
Plate1$Target[Plate1$Fluor == "VIC"] <- "scnag"
Plate1$Target[Plate1$Fluor == "FAM"] <- "mtdna"
Plate1$Target[Plate1$Fluor == "SYBR"] <- "telomeres"
```

Identify and remove outliers (across the three replicates)

```
# Add a column called "Diff_AVG-Cq"
Plate1$Diff_AVG_Cq <- abs(Plate1$Cq - Plate1$Cq.Mean)

## Add a column called "Flag_outlier"
Plate1$Flag_outlier <- ifelse(Plate1$Diff_AVG_Cq > 0.4001, "yes", "no")
HighCq <- Plate1[Plate1$Diff_AVG_Cq > "0.3001", ]
VeryHighCq <- Plate1[Plate1$Diff_AVG_Cq > "0.4001", ]

# Report number of rows to be removed
print(paste("Number of rows with High Cq:", nrow(VeryHighCq)))
```

```
## [1] "Number of rows with High Cq: 55"
```

```
# Make a table called  of the high cq values
HighCq_Samples<- HighCq[c("Well", "Sample", "Fluor", "Diff_AVG_Cq" )]
print(HighCq_Samples)
```

```
##     Well Sample Fluor Diff_AVG_Cq
## 1    A01   STD1   FAM   1.3847060
## 2    A02   STD1   FAM   0.8892797
## 3    A03   STD1   FAM   0.4954263
## 9    A09  4802B   FAM   0.3794761
## 10   A10          FAM         NaN
## 11   B01   STD2   FAM   0.4248667
## 12   B02   STD2   FAM   0.9176310
## 13   B03   STD2   FAM   0.4927643
## 20   B10          FAM         NaN
## 21   C01   STD3   FAM   0.5539893
## 23   C03   STD3   FAM   0.3088023
## 30   C10          FAM         NaN
## 31   D01   STD4   FAM   0.8053253
## 32   D02   STD4   FAM   1.0718804
## 40   E01   STD5   FAM   0.5539264
## 41   E02   STD5   FAM   0.5357873
## 58   G01   STD7   FAM   0.3187145
## 59   G02   STD7   FAM   0.4790690
## 67   H01    NTC   FAM         NaN
## 68   H02    NTC   FAM         NaN
## 69   H03    NTC   FAM         NaN
## 76   A01   STD1   VIC   1.4993430
## 77   A02   STD1   VIC   0.9717209
## 78   A03   STD1   VIC   0.5276221
## 85   A10          VIC         NaN
## 86   B01   STD2   VIC   0.4324728
## 87   B02   STD2   VIC   0.8929927
## 88   B03   STD2   VIC   0.4605199
## 95   B10          VIC         NaN
## 96   C01   STD3   VIC   0.5788151
## 98   C03   STD3   VIC   0.3903316
## 105  C10          VIC         NaN
## 106  D01   STD4   VIC   0.6940230
## 107  D02   STD4   VIC   1.0350892
## 108  D03   STD4   VIC   0.3410662
## 115  E01   STD5   VIC   0.5134086
## 116  E02   STD5   VIC   0.5199541
## 134  G02   STD7   VIC   0.4820915
## 142  H01    NTC   VIC         NaN
## 143  H02    NTC   VIC         NaN
## 144  H03    NTC   VIC         NaN
## 151  A01   STD1  SYBR   0.3712482
## 152  A02   STD1  SYBR   0.3824107
## 153  A03   STD1  SYBR   0.7536589
## 158  A08  4802B  SYBR   0.3662630
## 159  A09  4802B  SYBR   0.3344275
## 161  B01   STD2  SYBR   0.6920074
## 163  B03   STD2  SYBR   0.6148706
## 171  C01   STD3  SYBR   0.9051918
## 172  C02   STD3  SYBR   0.3736271
## 173  C03   STD3  SYBR   0.5315647
## 181  D01   STD4  SYBR   0.8224468
## 182  D02   STD4  SYBR   0.3069731
## 183  D03   STD4  SYBR   0.5154737
## 184  D04  4738B  SYBR   0.3300359
## 185  D05  4738B  SYBR   0.3319616
## 190  E01   STD5  SYBR   0.6974706
## 192  E03   STD5  SYBR   0.4641959
## 199  F01   STD6  SYBR   1.0065401
## 200  F02   STD6  SYBR   0.4351715
## 201  F03   STD6  SYBR   0.5713686
## 211  G04  4761B  SYBR   1.6367058
## 212  G05  4761B  SYBR   2.1022989
## 213  G06  4761B  SYBR   0.4655931
## 217  H01    NTC  SYBR   1.5223274
## 218  H02    NTC  SYBR   0.6796284
## 219  H03    NTC  SYBR   0.8426990
## 223  H07  4720B  SYBR   0.3440704
## 225  H09  4720B  SYBR   0.5518547
```

```
# Filter and remove rows where Diff_AVG_Cq > 0.4001 for each SampleID
# USed to be max(AVG) but error said that col did not exist
dim(Plate1)
```

```
## [1] 225  19
```

```
Plate1 <- Plate1 %>%
  group_by(Sample, Target) %>%
  filter(!(Diff_AVG_Cq > 0.4001 & Diff_AVG_Cq == max(Diff_AVG_Cq)))
dim(Plate1)
```

```
## [1] 192  19
```

Additional identification and removal of outliers

```
# Recalculate Cq Mean and Add a column called "Cq Mean2"
Plate1 <- Plate1 %>%
  group_by(Sample, Target) %>%
  mutate(
    Cq.Mean2 = mean(Cq),
    SQ.Mean2 = mean(Starting.Quantity..SQ.)) %>%
  ungroup()
dim(Plate1)
```

```
## [1] 192  21
```

```
# Add a column called "Diff_AVG-Cq_2" that contains the difference between new Cq and new mean
Plate1$Diff_AVG_Cq_2 <- abs(Plate1$Cq - Plate1$Cq.Mean2)

# Add a column called "Flag_outlier_2"
Plate1$Flag_outlier_2 <- ifelse(Plate1$Diff_AVG_Cq_2 > 0.401, "yes", "no")
VeryHighCq_2 <- Plate1[Plate1$Diff_AVG_Cq > "0.401", ]
# Report number of rows to be removed
print(paste("Number of rows with High Cq:", nrow(VeryHighCq_2)))
```

```
## [1] "Number of rows with High Cq: 22"
```

```r
# Make a table called  of the high cq values
HighCq_2_Samples<- HighCq[c("Well", "Sample", "Fluor", "Diff_AVG_Cq" )]
print(HighCq_2_Samples)
```

```
##     Well Sample Fluor Diff_AVG_Cq
## 1    A01   STD1   FAM   1.3847060
## 2    A02   STD1   FAM   0.8892797
## 3    A03   STD1   FAM   0.4954263
## 9    A09  4802B   FAM   0.3794761
## 10   A10          FAM         NaN
## 11   B01   STD2   FAM   0.4248667
## 12   B02   STD2   FAM   0.9176310
## 13   B03   STD2   FAM   0.4927643
## 20   B10          FAM         NaN
## 21   C01   STD3   FAM   0.5539893
## 23   C03   STD3   FAM   0.3088023
## 30   C10          FAM         NaN
## 31   D01   STD4   FAM   0.8053253
## 32   D02   STD4   FAM   1.0718804
## 40   E01   STD5   FAM   0.5539264
## 41   E02   STD5   FAM   0.5357873
## 58   G01   STD7   FAM   0.3187145
## 59   G02   STD7   FAM   0.4790690
## 67   H01    NTC   FAM         NaN
## 68   H02    NTC   FAM         NaN
## 69   H03    NTC   FAM         NaN
## 76   A01   STD1   VIC   1.4993430
## 77   A02   STD1   VIC   0.9717209
## 78   A03   STD1   VIC   0.5276221
## 85   A10          VIC         NaN
## 86   B01   STD2   VIC   0.4324728
## 87   B02   STD2   VIC   0.8929927
## 88   B03   STD2   VIC   0.4605199
## 95   B10          VIC         NaN
## 96   C01   STD3   VIC   0.5788151
## 98   C03   STD3   VIC   0.3903316
## 105  C10          VIC         NaN
## 106  D01   STD4   VIC   0.6940230
## 107  D02   STD4   VIC   1.0350892
## 108  D03   STD4   VIC   0.3410662
## 115  E01   STD5   VIC   0.5134086
## 116  E02   STD5   VIC   0.5199541
## 134  G02   STD7   VIC   0.4820915
## 142  H01    NTC   VIC         NaN
## 143  H02    NTC   VIC         NaN
## 144  H03    NTC   VIC         NaN
## 151  A01   STD1  SYBR   0.3712482
## 152  A02   STD1  SYBR   0.3824107
## 153  A03   STD1  SYBR   0.7536589
## 158  A08  4802B  SYBR   0.3662630
## 159  A09  4802B  SYBR   0.3344275
## 161  B01   STD2  SYBR   0.6920074
## 163  B03   STD2  SYBR   0.6148706
## 171  C01   STD3  SYBR   0.9051918
## 172  C02   STD3  SYBR   0.3736271
## 173  C03   STD3  SYBR   0.5315647
## 181  D01   STD4  SYBR   0.8224468
## 182  D02   STD4  SYBR   0.3069731
## 183  D03   STD4  SYBR   0.5154737
## 184  D04  4738B  SYBR   0.3300359
## 185  D05  4738B  SYBR   0.3319616
## 190  E01   STD5  SYBR   0.6974706
## 192  E03   STD5  SYBR   0.4641959
## 199  F01   STD6  SYBR   1.0065401
## 200  F02   STD6  SYBR   0.4351715
## 201  F03   STD6  SYBR   0.5713686
## 211  G04  4761B  SYBR   1.6367058
## 212  G05  4761B  SYBR   2.1022989
## 213  G06  4761B  SYBR   0.4655931
## 217  H01    NTC  SYBR   1.5223274
## 218  H02    NTC  SYBR   0.6796284
## 219  H03    NTC  SYBR   0.8426990
## 223  H07  4720B  SYBR   0.3440704
## 225  H09  4720B  SYBR   0.5518547
```

```
# Remove the rows with a high Cq Outliers
# USed to be max(AVG) but error said that col did not exist

Plate1 <- Plate1 %>%
  group_by(Sample, Target) %>%
  filter(!(Diff_AVG_Cq_2 > 0.4001 & Diff_AVG_Cq_2 == max(Diff_AVG_Cq_2)))

dim(Plate1)
```

```
## [1] 189  23
```

```
###############  If any "Samples" do not have more than two rows (replicates left), remove
Plate1 <- Plate1 %>%
  group_by(Sample, Target) %>%
  filter(n() >= 2) %>%
  ungroup()
dim(Plate1)
```

```
## [1] 188  23
```

Remove negative controls and standards

```
# rows that have "NEG", "POS" in column "Sample" and remove rows with "STD" in Sample "Content"
Plate1 <- Plate1 %>%
  filter(!str_detect(Content, "Std-*")) %>%
  filter(!str_detect(Content, "NTC")) %>%
  filter(!str_detect(Sample, "Neg")) %>%
  filter(!str_detect(Sample, "STD*"))
dim(Plate1)
```

```
## [1] 140  23
```

Subset dataset based on the value in “Target” column

```
unique_targets <- unique(Plate1$Target)
# Create a list to store the subset dataframes
subset_dfs <- list()
# Loop through each unique value in 'Target', subset the dataframe, and store in subset_dfs
for (target_value in unique_targets) {
  subset_df <- subset(Plate1, Target == target_value)
  subset_dfs[[target_value]] <- subset_df
}
```

Now subset_dfs is a list where each element is a dataframe containing rows for each unique ‘Target’ value

```
Plate1_SCNAG<-print(subset_dfs[["scnag"]])
```

```
## # A tibble: 48 x 23
##    X     Well  Fluor Target Content Sample Biological.Set.Name    Cq Cq.Mean
##    <lgl> <chr> <chr> <chr>  <chr>   <chr>  <lgl>               <dbl>   <dbl>
##  1 NA    A04   VIC   scnag  Unkn-01 4775B  NA                   23.8    23.8
##  2 NA    A05   VIC   scnag  Unkn-01 4775B  NA                   23.9    23.8
##  3 NA    A06   VIC   scnag  Unkn-01 4775B  NA                   23.8    23.8
##  4 NA    A07   VIC   scnag  Unkn-09 4802B  NA                   23.8    23.7
##  5 NA    A08   VIC   scnag  Unkn-09 4802B  NA                   23.8    23.7
##  6 NA    A09   VIC   scnag  Unkn-09 4802B  NA                   23.6    23.7
##  7 NA    B04   VIC   scnag  Unkn-02 4792B  NA                   24.6    24.7
##  8 NA    B05   VIC   scnag  Unkn-02 4792B  NA                   24.6    24.7
##  9 NA    B06   VIC   scnag  Unkn-02 4792B  NA                   24.8    24.7
## 10 NA    B07   VIC   scnag  Unkn-10 4739B  NA                   25.0    25.0
## # i 38 more rows
## # i 14 more variables: Cq.Std..Dev <dbl>, Starting.Quantity..SQ. <dbl>,
## #   Log.Starting.Quantity <dbl>, SQ.Mean <dbl>, SQ.Std..Dev <dbl>,
## #   Set.Point <int>, Well.Note <lgl>, PlateID <chr>, Diff_AVG_Cq <dbl>,
## #   Flag_outlier <chr>, Cq.Mean2 <dbl>, SQ.Mean2 <dbl>, Diff_AVG_Cq_2 <dbl>,
## #   Flag_outlier_2 <chr>
```

```
Plate1_SCNAG<-Plate1_SCNAG[ ,c("PlateID", "Well", "Sample", "Target", "Cq", "Cq.Mean",  "Flag_outlier",
Plate1_SCNAG <- Plate1_SCNAG %>%
  rename(Target_SCNAG = Target, Cq_SCNAG = Cq, Cq.Mean_SCNAG = Cq.Mean, Flag_outlier_SCNAG=Flag_outlier

Plate1_mtDNA<-print(subset_dfs[["mtdna"]])
```

```
## # A tibble: 48 x 23
##    X     Well  Fluor Target Content Sample Biological.Set.Name    Cq Cq.Mean
##    <lgl> <chr> <chr> <chr>  <chr>   <chr>  <lgl>               <dbl>   <dbl>
##   1 NA    A04   FAM   mtdna  Unkn-01 4775B  NA                    27.3    27.4
##   2 NA    A05   FAM   mtdna  Unkn-01 4775B  NA                    27.5    27.4
##   3 NA    A06   FAM   mtdna  Unkn-01 4775B  NA                    27.3    27.4
##   4 NA    A07   FAM   mtdna  Unkn-09 4802B  NA                    26.2    26.0
##   5 NA    A08   FAM   mtdna  Unkn-09 4802B  NA                    26.2    26.0
##   6 NA    A09   FAM   mtdna  Unkn-09 4802B  NA                    25.6    26.0
##   7 NA    B04   FAM   mtdna  Unkn-02 4792B  NA                    26.2    26.3
##   8 NA    B05   FAM   mtdna  Unkn-02 4792B  NA                    26.1    26.3
##   9 NA    B06   FAM   mtdna  Unkn-02 4792B  NA                    26.4    26.3
##  10 NA    B07   FAM   mtdna  Unkn-10 4739B  NA                    26.5    26.6
## # i 38 more rows
## # i 14 more variables: Cq.Std..Dev <dbl>, Starting.Quantity..SQ. <dbl>,
## #   Log.Starting.Quantity <dbl>, SQ.Mean <dbl>, SQ.Std..Dev <dbl>,
## #   Set.Point <int>, Well.Note <lgl>, PlateID <chr>, Diff_AVG_Cq <dbl>,
## #   Flag_outlier <chr>, Cq.Mean2 <dbl>, SQ.Mean2 <dbl>, Diff_AVG_Cq_2 <dbl>,
## #   Flag_outlier_2 <chr>
```

```
Plate1_mtDNA<-Plate1_mtDNA[ ,c("PlateID", "Well", "Sample", "Target", "Cq", "Cq.Mean", "Flag_outlier",
Plate1_mtDNA <- Plate1_mtDNA %>%
  rename(Target_mtDNA = Target, Cq_mtDNA = Cq, Cq.Mean_mtDNA = Cq.Mean, Flag_outlier_mtDNA = Flag_outli

Plate1_Telomeres<-print(subset_dfs[["telomeres"]])
```

```
## # A tibble: 44 x 23
##    X     Well  Fluor Target    Content Sample Biological.Set.Name    Cq Cq.Mean
##    <lgl> <chr> <chr> <chr>     <chr>   <chr>  <lgl>               <dbl>   <dbl>
##   1 NA    A04   SYBR  telomeres Unkn-01 4775B  NA                    15.3    15.3
##   2 NA    A05   SYBR  telomeres Unkn-01 4775B  NA                    15.2    15.3
##   3 NA    A06   SYBR  telomeres Unkn-01 4775B  NA                    15.3    15.3
##   4 NA    A07   SYBR  telomeres Unkn-09 4802B  NA                    15.8    15.8
##   5 NA    A08   SYBR  telomeres Unkn-09 4802B  NA                    15.4    15.8
##   6 NA    A09   SYBR  telomeres Unkn-09 4802B  NA                    16.1    15.8
##   7 NA    B04   SYBR  telomeres Unkn-02 4792B  NA                    15.5    15.7
##   8 NA    B05   SYBR  telomeres Unkn-02 4792B  NA                    15.8    15.7
##   9 NA    B06   SYBR  telomeres Unkn-02 4792B  NA                    15.7    15.7
##  10 NA    B07   SYBR  telomeres Unkn-10 4739B  NA                    15.4    15.4
## # i 34 more rows
## # i 14 more variables: Cq.Std..Dev <dbl>, Starting.Quantity..SQ. <dbl>,
## #   Log.Starting.Quantity <dbl>, SQ.Mean <dbl>, SQ.Std..Dev <dbl>,
## #   Set.Point <int>, Well.Note <lgl>, PlateID <chr>, Diff_AVG_Cq <dbl>,
## #   Flag_outlier <chr>, Cq.Mean2 <dbl>, SQ.Mean2 <dbl>, Diff_AVG_Cq_2 <dbl>,
## #   Flag_outlier_2 <chr>
```

```
Plate1_Telomeres<-Plate1_Telomeres[ ,c("PlateID", "Well", "Sample", "Target", "Cq", "Cq.Mean", "Flag_ou
Plate1_Telomeres <- Plate1_Telomeres %>%
  rename(Cq_Telomeres = Cq, Cq.Mean_Telomeres = Cq.Mean, Flag_outlier_Telomeres = Flag_outlier, SQ.Mean_
```

Make a final MPX dataset for by merging the Target datasets horizontally, in rows.

```
Plate1_FinalMPX <- merge(Plate1_SCNAG, Plate1_mtDNA, by = c("PlateID", "Well", "Sample"))

# Normalize mtDNA
# Add a column called mtDNA, and calculate the normalized value
Plate1_FinalMPX$mtDNA <- (Plate1_FinalMPX$SQ.Mean_mtDNA / Plate1_FinalMPX$SQ.Mean_SCNAG)

# Recalculate mean across the replicates
Plate1_FinalMPX <- Plate1_FinalMPX %>%
  group_by(Sample) %>%
  mutate(
    mtDNA.Mean = mean(mtDNA)) %>%
  ungroup()
```

Merge final MPX with Telomeres

```
## Reduce datasets to single row per individual containing only the columns we want.
Plate1_FinalMPX <- distinct(Plate1_FinalMPX, PlateID, Sample, SQ.Mean_SCNAG, Cq.Mean_SCNAG, SQ.Mean_mtD
Plate1_FinalTelo <- distinct(Plate1_Telomeres, PlateID, Sample, SQ.Mean_Telomeres, Cq.Mean_Telomeres)

## Merge the files horizontally
Plate1_FinalData <- merge(Plate1_FinalMPX, Plate1_FinalTelo, by = c("PlateID", "Sample"))

# Normalize Telomeres
Plate1_FinalData <- Plate1_FinalData %>% mutate(Telomeres.per.cell = SQ.Mean_Telomeres / SQ.Mean_SCNAG)
```

Aggregate to get one row per sample, taking mean of normalized mtDNA and telomeres

```
Plate1_FinalData <- Plate1_FinalData %>%
  group_by(Sample) %>%
  summarize(
    SQ.Mean_SCNAG = mean(SQ.Mean_SCNAG),
    SQ.Mean_mtDNA = mean(SQ.Mean_mtDNA),
    mtDNA.Mean = mean(mtDNA.Mean),
    Cq.Mean_SCNAG = mean(Cq.Mean_SCNAG),
    SQ.Mean_Telomeres = mean(SQ.Mean_Telomeres),
    Cq.Mean_Telomeres = mean(Cq.Mean_Telomeres),
    Telomeres.per.cell = mean(Telomeres.per.cell)
  ) %>%
  ungroup()
```

Merge Final Data with Trait MetaData for your individuals

```
# Load in Data
Trait <- read.csv("Trait_MetaData.csv")
dim (Trait)
```

```
## [1] 16  8
```

```
# Merge both datasets
FinalData <- merge(Plate1_FinalData, Trait, by = c("Sample"))
```

Write the final data file for this plate

```
write.csv(file = "AHDB1_Manuscript1_ESEBqPCR_FinalData.csv", FinalData, row.names = FALSE)
```

Data analysis

Load data

```
datum <- read.csv("AHDB1_Manuscript1_ESEBqPCR_FinalData.csv")
head(datum)
```

```
##   Sample SQ.Mean_SCNAG SQ.Mean_mtDNA mtDNA.Mean Cq.Mean_SCNAG SQ.Mean_Telomeres
## 1  4625B     974.06273      63.21475 0.06489803      21.69972         353530.23
## 2  4720B     165.41706     128.70042 0.77803592      24.19144         126633.25
## 3  4737B     204.94218      30.21435 0.14742865      23.89250          90483.14
## 4  4738B     123.49941      26.62164 0.21556089      24.60234          54701.82
## 5  4739B      92.00273      48.29679 0.52494951      25.02672         149955.61
## 6  4745B     175.26504      54.82246 0.31279748      24.11047          95191.94
##   Cq.Mean_Telomeres Telomeres.per.cell AgeCategory   Sex Mom_ID Cohort
## 1          13.93421           362.9440     Younger FALSE   4450      4
## 2          15.67509           765.5392     Younger FALSE   4374      5
## 3          16.19194           441.5057     Younger FALSE   4210      6
## 4          17.04050           442.9318     Younger FALSE   4210      6
## 5          15.35503          1629.9039     Younger FALSE   4023      5
## 6          16.10705           543.1313     Younger FALSE   4111      6
##   Treatment Died_InTrt Time_Point
## 1         E         No   Baseline
## 2         E         No   Baseline
## 3         C        Yes   Baseline
## 4         E         No   Baseline
## 5         E         No   Baseline
## 6         E         No   Baseline
```

Convert variables to factors

```
datum$Sample <- as.factor(datum$Sample)
datum$Died_InTrt <- as.factor(datum$Died_InTrt)

str(datum)
```

```
## 'data.frame':    15 obs. of  15 variables:
##  $ Sample            : Factor w/ 15 levels "4625B","4720B",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ SQ.Mean_SCNAG     : num  974 165 205 123 92 ...
##  $ SQ.Mean_mtDNA     : num  63.2 128.7 30.2 26.6 48.3 ...
##  $ mtDNA.Mean        : num  0.0649 0.778 0.1474 0.2156 0.5249 ...
##  $ Cq.Mean_SCNAG     : num  21.7 24.2 23.9 24.6 25 ...
##  $ SQ.Mean_Telomeres : num  353530 126633 90483 54702 149956 ...
##  $ Cq.Mean_Telomeres : num  13.9 15.7 16.2 17 15.4 ...
##  $ Telomeres.per.cell: num  363 766 442 443 1630 ...
##  $ AgeCategory       : chr  "Younger" "Younger" "Younger" "Younger" ...
##  $ Sex               : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Mom_ID            : int  4450 4374 4210 4210 4023 4111 4147 4171 4171 4127 ...
##  $ Cohort            : int  4 5 6 6 5 6 7 7 7 8 ...
##  $ Treatment         : chr  "E" "E" "C" "E" ...
##  $ Died_InTrt        : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 2 2 1 ...
##  $ Time_Point        : chr  "Baseline" "Baseline" "Baseline" "Baseline" ...
```

Hypothesis 3. Among young adult females, there will be metabolic or damage variables in the baseline blood samples (2 weeks prior) may correlated with probability of dying due to acute heat (~43 – 43.5C for 5 hours).

GLMER -> mtDNA Copy Number Model (WITH LYSED BUT NO Mom_ID)

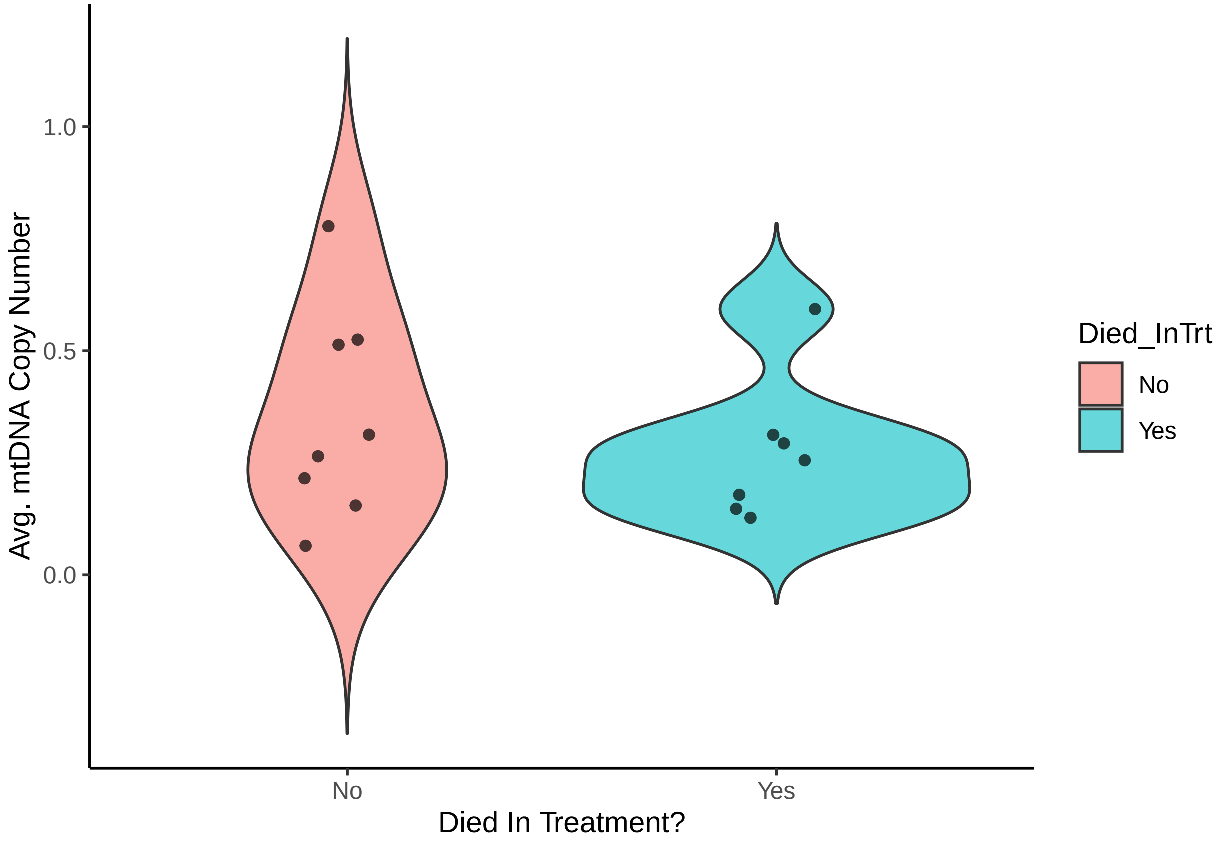
```
glmer_model_Mito <- glmer(Died_InTrt ~  mtDNA.Mean +
                            (1 | Cohort),
                          data = datum,
                          family = binomial)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(glmer_model_Mito)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Died_InTrt ~ mtDNA.Mean + (1 | Cohort)
##    Data: datum
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##      26.1      28.2     -10.0      20.1        12
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.2398 -0.9593 -0.5468  0.9616  1.4790
##
## Random effects:
##  Groups Name        Variance  Std.Dev.
##  Cohort (Intercept) 3.868e-15 6.219e-08
## Number of obs: 15, groups:  Cohort, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5788     1.0256   0.564    0.572
## mtDNA.Mean   -2.2955     2.9029  -0.791    0.429
##
## Correlation of Fixed Effects:
##            (Intr)
## mtDNA.Mean -0.857
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
# Results not significant (p=0.429)

# Plot
ggplot(datum, aes(x = Died_InTrt, y = mtDNA.Mean)) +
  geom_violin(aes(fill = Died_InTrt), trim = FALSE, alpha = 0.6) +  # Violin by group
  geom_jitter(width = 0.1, alpha = 0.7, size = 1.5, color = "black") +  # Add data points
  labs(
    x = "Died In Treatment?",
    y = "Avg. mtDNA Copy Number"
  ) +
  theme_classic()
```



GLMER -> Telomere (T/S ratio) Model (WITH LYSED BUT NO Mom_ID)

```
glmer_model_Telo <- glmer(Died_InTrt ~  Telomeres.per.cell +
                            (1 | Cohort),
                         data = datum,
                         family = binomial)
```

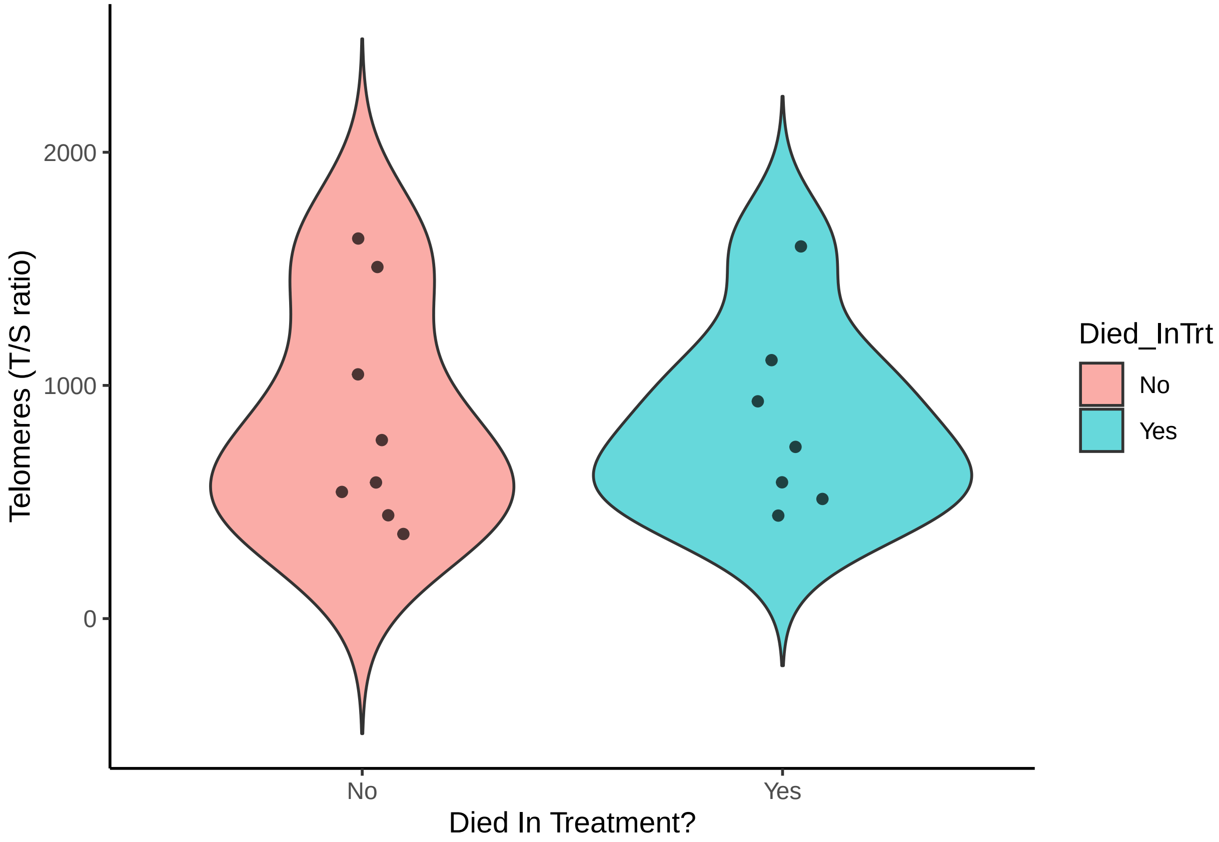
```
## boundary (singular) fit: see help('isSingular')
```

```
summary(glmer_model_Telo)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Died_InTrt ~ Telomeres.per.cell + (1 | Cohort)
##    Data: datum
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##      26.7      28.8     -10.4      20.7        12
##
```

```
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -0.9562 -0.9429 -0.9033  1.0599  1.1054 
## 
## Random effects:
##  Groups Name        Variance  Std.Dev.
##  Cohort (Intercept) 1.406e-15 3.75e-08
## Number of obs: 15, groups:  Cohort, 7
## 
## Fixed effects:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -5.695e-02  1.168e+00  -0.049    0.961
## Telomeres.per.cell -8.984e-05  1.230e-03  -0.073    0.942
## 
## Correlation of Fixed Effects:
##             (Intr)
## Tlmrs.pr.cl -0.896
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
# Results not significant (p=0.942)

# Plot
ggplot(datum, aes(x = Died_InTrt, y = Telomeres.per.cell)) +
  geom_violin(aes(fill = Died_InTrt), trim = FALSE, alpha = 0.6) +  # Violin by group
  geom_jitter(width = 0.1, alpha = 0.7, size = 1.5, color = "black") +  # Add data points
  labs(
    x = "Died In Treatment?",
    y = "Telomeres (T/S ratio)"
  ) +
  theme_classic()
```



Double checking SCNAG

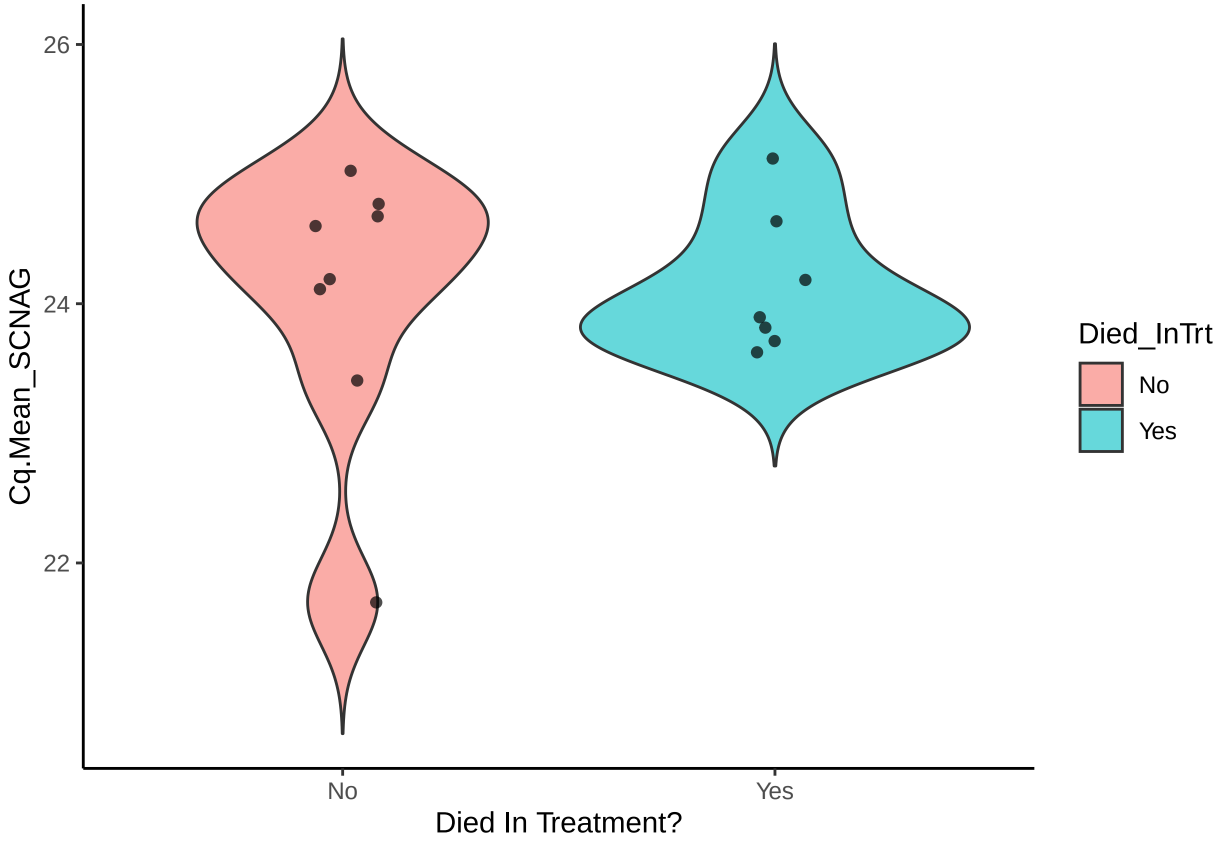
```
glmer_model_SCNAG <- glmer(Died_InTrt ~  Cq.Mean_SCNAG +
                            (1 | Cohort),
                         data = datum,
                         family = binomial)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(glmer_model_SCNAG)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Died_InTrt ~ Cq.Mean_SCNAG + (1 | Cohort)
##    Data: datum
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##      26.7      28.8     -10.3      20.7        12
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.9903 -0.9527 -0.8067  1.0734  1.1006
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Cohort (Intercept) 0        0
## Number of obs: 15, groups:  Cohort, 7
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.1047    15.4705  -0.201    0.841
## Cq.Mean_SCNAG   0.1233     0.6414   0.192    0.848
##
## Correlation of Fixed Effects:
##             (Intr)
## Cq.Mn_SCNAG -0.999
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
# Results not significant (p=0.848)

# Plot
ggplot(datum, aes(x = Died_InTrt, y = Cq.Mean_SCNAG)) +
  geom_violin(aes(fill = Died_InTrt), trim = FALSE, alpha = 0.6) +  # Violin by group
  geom_jitter(width = 0.1, alpha = 0.7, size = 1.5, color = "black") +  # Add data points
  labs(
    x = "Died In Treatment?",
    y = "Cq.Mean_SCNAG"
  ) +
  theme_classic()
```



ESEB Poster Brynleigh -Graph of Glucose, Ketone, Hematocrit averages

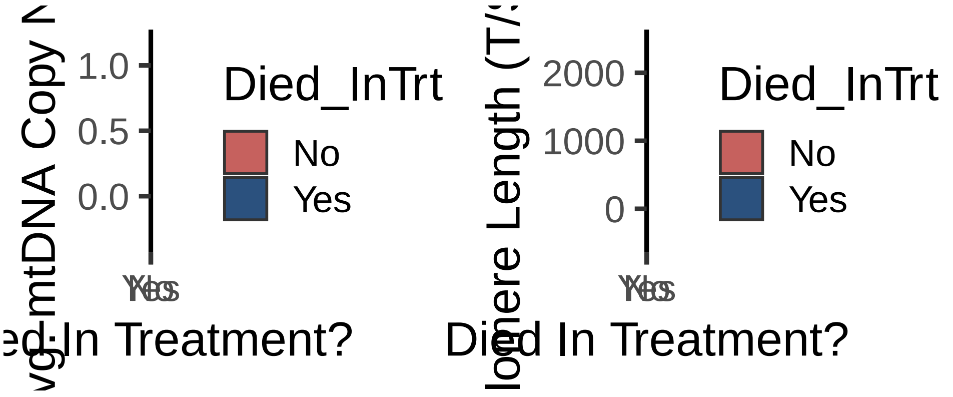
Models do not include mom ID due to only two being the same within the died in treatment individuals

Includes cohort as a random effect

```
# Mitochondrial Plot
p1 <- ggplot(datum, aes(x = Died_InTrt, y = mtDNA.Mean)) +
  geom_violin(aes(fill = Died_InTrt), trim = FALSE, alpha = 0.9) +  # Violin by group
  geom_jitter(width = 0.1, alpha = 0.7, size = 1.5, color = "black") +  # Add data points
  scale_fill_manual(values = c("#C0504D", "#153F70")) +  # Custom colors
  labs(
    x = "Died In Treatment?",
    y = "Avg. mtDNA Copy Number"
  ) +
  theme_classic(base_size = 18) +
  theme()  # Removes the legend

# Telomere Plot
p2 <- ggplot(datum, aes(x = Died_InTrt, y = Telomeres.per.cell)) +
  geom_violin(aes(fill = Died_InTrt), trim = FALSE, alpha = 0.9) +  # Violin by group
  geom_jitter(width = 0.1, alpha = 0.7, size = 1.5, color = "black") +  # Add data points
  scale_fill_manual(values = c("#C0504D", "#153F70")) +  # Custom colors
  labs(
    x = "Died In Treatment?",
    y = "Telomere Length (T/S Ratio)"
  ) +
  theme_classic(base_size = 18) +
  theme()  # Removes the legend

library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
# Combine into 1 row and 3 columns
combined_qPCR_plot <- plot_grid(p1, p2, nrow = 1)
combined_qPCR_plot
```



```
#ggsave("Combined_qPCR_Plot.png", combined_qPCR_plot, width = 12, height = 4)
```